

Übungen zur Einführung in die Programmiersprache Java

Universität Regensburg
NWF II - Physik
Dominik Köppl

Wintersemester 2011/12
Blatt 10

36 Suche in Dateien

Sie haben eine Datei gespeichert, wissen aber nicht mehr ihren Namen. Deswegen wollen Sie ein Programm entwerfen, welches nach einer Zeichenkette innerhalb einer bestimmten Datei Ausschau hält. Dazu deklarieren Sie die Funktion `int sucheString(String dateiname, String suchtext)`. Zunächst öffnet die Funktion die Datei `dateiname` und beginnt, die Datei nach dem `suchtext` komplett zu durchsuchen. Nutzen Sie dabei einen `BufferedReader`, um die Datei zeilenweise auszulesen. Für jede Zeile können Sie mit einer doppelten Schleife und der `charAt`-Methode für Strings nach dem Suchtext buchstabenweise suchen. Dabei zählt die Funktion die Anzahl der Treffer. Konnte die Datei nicht geöffnet werden, wirft die Funktion die Exception `java.io.FileNotFoundException`.

37 Rechtecksintegration

Sicherlich ist Ihnen noch die Einführung der Integration aus der Schule bekannt: Damals hatten Sie die Integration mit Rechtecken approximiert, die die Fläche des Graphen $\Gamma : x \mapsto (x, f(x))$ mit $f : (a, b) \rightarrow \mathbb{R}$ ausfüllen sollen (wobei $a < b \in \mathbb{R}$). Je mehr Rechtecke Sie verwenden, desto kleiner werden ihre Breiten und desto genauer spiegeln diese die Fläche des Graphen wider.

- Man unterteile das Intervall (a, b) in n Teilstücke, s.d. wir eine Aufteilung $a = t_0 < t_1 < \dots < t_n = b$ erhalten.
- Das Rechteck R_i ist dann durch die Punkte $(t_i, f(t_i))$ und $(t_{i+1}, f(t_i))$ für alle $i = 0, \dots, n - 1$ gegeben.
- Zählt man nun den Flächeninhalt aller Rechtecke zusammen, erhält man eine Approximation des Integrals von f über das Intervall (a, b) .

Wir werden das Ganze nun in Java implementieren:

- (a) Schreiben Sie ein Interface `Integrable`, welches eine abstrakte Methode `double function(double x)`; und zwei Getter `double getA()`; und `double getB()`; bereitstellt.
- (b) Implementieren Sie dieses Interface in zwei von Ihnen selbstdefinierten Klassen (z.B. können Sie mit `function` den Cosinus berechnen lassen). Sie sollten zu Testzwecken nur Funktionen verwenden, von denen Sie das unbestimmte Integral kennen!
- (c) Anschließend schreiben Sie sich eine Klasse, die die Integrationsmethode `double integrate(Integrable integr, int n)` bereitstellt.

- (d) Diese Methode soll nun das Intervall (`getA`, `getB`) in n -Teile aufteilen und mit Hilfe der Rechtecksintegration die Integration approximieren.

38 Bankdaten

Sie sind als Bankkauffrau/-mann in einem Unternehmen tätig. Dazu müssen Sie oft Sammelüberweisungen tätigen. Die Überweisungen müssen Sie jedoch zuerst in eine Datei abspeichern. Dafür hat Ihnen das Unternehmen folgendes Dateischema auferlegt:

- Eine Überweisung ist von der Klasse `Transaction`
 - In die Datei werden einfach nacheinander die `Transaction`en abgespeichert.
 - Die Daten werden im Binärformat abgespeichert.
 - Am Anfang der Datei steht eine Zahl, wieviele Überweisungen schon abgespeichert wurden.
- (a) Entwerfen Sie die Klasse `Transaction`. Diese Klasse soll die Attribute `Kontoinhaber`, `Kontonummer`, `Bankleitzahl` und `Betrag` beinhalten. Nutzen Sie passende Datentypen für die Attribute. Schreiben Sie einen vollständigen Konvertierungskonstruktor.
- (b) Implementieren Sie das Interface `Externalizable`. Denken Sie daran, dass die Daten im Binärformat ein- und ausgelesen werden sollen.
- (c) In der `main`-Routine soll zunächst versucht werden, die Kontodatei zu öffnen um dann den Inhalt auszugeben. Dazu sollten Sie die `toString()`-Methode verwenden. Anschließend fragt das Programm, ob Sie eine erneute Überweisung tätigen wollen. Falls ja, wird diese zu den übrigen Überweisungen in die Datei hinzugefügt.

Hinweis Nutzen Sie die Zahl am Anfang der Datei, um ein Array aus `Transaction`en der gewünschten Größe erzeugen zu können.

39 Bonusaufgabe: Komplexe Zahlen

Komplexe Zahlen werden oft für Berechnungen benötigt. Anders wie in C/C++ gibt es in Java keine Klasse für komplexe Zahlen!

- (a) Daher erstellen wir uns eine eigene Klasse `Complex`, die das Interface `Field<Complex>` implementiert. Die Klasse beherbergt zwei Attribute des Types `double`: Den Realteil und den Imaginärteil. Man identifiziert $\mathbb{C} \simeq \mathbb{R}^2$, $(a + ib) \mapsto (a, b)$, wobei $\sqrt{-1} = i \in \mathbb{C}$. Zusätzlich soll die Klasse die Funktion `Complex sqrt(double a)` bereitstellen, die Wurzeln von positiven und negativen Zahlen berechnen kann.
- (b) Berechnen Sie erneut die Mitternachtsformel, wobei Sie in allen Fällen eine Lösung zurückgeben können.